Modeling Cometary Coma with a Three Dimensional, Anisotropic Multiple Scattering Distributed Processing Code

Chris B. Luchini[a] *

[a]Jet Propulsion Laboratory,
4800 Oak Grove Drive, Pasadena, California 91109

Development of camera and instrument simulations for space exploration requires the development of scientifically accurate models of the objects to be studied. Several planned cometary missions have prompted the development of a three dimensional, multi-spectral, anisotropic multiple scattering model of cometary coma. Pre-existing single processor codes have been adapted to multi-threaded and distributed processing implementations.

1. Comet coma illumination modeling

Modeling of spacecraft imaging systems for cometary missions requires a high fidelity illumination and luminosity model of the cometary coma. The light distribution and luminosity of cometary coma is complex problem that is not adequately addressed by conventional isotropic radiation transfer techniques. The optical cross sections due to dust and gas concentration vary by 4 or more orders of magnitude across the spatial region of interest. The optical density has significant small scale structure near the compact dust and gas jets that are emitted from the cometary nucleus. The length scale of interest varies from ~ 10^5 meters far from comet nucleus to of order 10 meters at the surface of the nucleus. The sunlight scattered from the dust is highly non-isotropic in distribution. These characteristics have prompted the development of an illumination, light distribution and rendering code that is fully three- dimensional, incorporating anisotropic multiple scattering, and progressive refinement of the computational grid.

1.1. Radiosity solutions

In general volume rendering of attenuating, non-isotropic scattering objects is quite computationally expensive. While analytic radiosity methods for non-transparent objects and environments are fairly common, the computational cost of solving a (N voxel x M direction bin)^2 element (albeit sparse) matrix for reasonable spatial and directional discretization restricts these solutions to small problems. Iterative Monte Carlo radiosity solutions to the full multiple scattering problem allow a trade off of computational

*This work was carried out for Model based Planetary Data Analysis Task at the Jet Propulsion Laboratory, California Institute of Technology, under the contract with the National Aeronautics and Space Administration. The funding was provided by the Information Systems Branch of Code STI/NASA and the office of Mission to Planet Earth as a part of NASA's High Performance Computing and Communication program.

resources and precision.

In one of the more common problems addressed in the literature, that of clouds, the albedo is very high $\simeq 0.99$ requiring many multiple scattering passes be made for adequate accuracy of the filial light distribution. Various methods have been developed to reduce the cost of these Monte Carlo multiple scattering calculations[1][3], however they tend to suffer from "beaming" or other artifacts[3]. Since the low albedo of the comet dust $\simeq 0.04$[6] means that any scattered light is much more likely to be absorbed that to undergo an additional scattering, the low albedo coma requires many fewer multiple scattering iterations for a acceptable level of accuracy. A straight forward Monte Carlo approach with a semi-analytic light interaction, attenuation, and directional scattering model gives a scientifically valid visualization without the known artifacts of the afore mentioned approaches.

2. Fully three dimensional model of the coma

Previous comet models have generally been cylindrically or spherically symmetric. These models have been quite successful for the extended coma at large distances from the comet nucleus. However, comets are believed to emit a large fraction of their dust and gas via compact jets emitted from the sun-ward side of the nucleus. These jets are quite dense compared to the expanded coma, and may include relatively high albedo (~ 0.5 or more) water ice. In general these jets are though to be at fixed positions on the comet nucleus surface, however the tumbling comet nucleus rotates these jets in and out of th sunlight, turning them on and off with each rotation. As these jets move off into space, they produce significant structure in the far coma and tail of the comet. Modeling of the light from compact dust and gas jets emitted from the comet nucleus requires a full three dimensional spatial grid.

3. Computational structure

The generated dust and gas distributions are binned in a Cartesian grid of cubic volume elements (voxels). Progressive refinement of the grid is used to achieve the increased spatial resolution near the nucleus needed to resolve nuclear dust and gas jets, as well as to adapt to the steeply falling dust density ($r^{-2.6}$)[4] as a function of distance from the comet nucleus.

The cometary dust has a very non-isotropic visible light scattering phase function. To keep track of the direction of the scattered light, each volume element is directionally discritized into N x N angular bins. The number of direction bins may be varied for voxels across the grid, as low dust density regions, and/or regions that received little scattered light may be described with fewer direction bins. For regions where multiple scattering is considered to be unimportant, the directional discretization may be replaced by an analytic function describing the light emission direction from the voxel.

Physical models of the time evolution of the ejected dust indicate that dust particles decay to smaller sizes[5]. It is expected that the phase angle of light scattered from the larger particles near the cometary nucleus will be substantially different that the scattering phase distribution of the light scattered from smaller particles far from the nucleus. Each volume element can have an individual scattering phase function, selected from a set of analytic phase functions.

For the purposes of this paper a single scattering phase function was used for the entire grid, and the directional discretization was fixed at 100 direction bins/voxel.

4. Multiple Scattering

Unlike multiple scattering in high albedo, high optical thickness objects like clouds, multiple scattering in a cometary dust coma is a minor effect. The average albedo of comet dust is of order $a \simeq 0.04$, with significant increases from this value found only in dust and gas jets near the nucleus. The solar-nucleus optical depth ω_{sc} is generally less than 1. The additional precision gained from each multiple scattering simulation is of orde1('1' $a \cdot \omega_{sc} << 0.04$. Thus first order Monte Carlo scattering is the highest order that is normally done, though given sufficient resources any order of multiple scattering can be computed. Though the light distribution in the comet coma is an off line (non-time critical) calculation, an parallel processing version of this code was developed in order to shorten the typical 20 hour single processor calculation time for each Monte Carlo scattering step.

The multiple scattering solution is generated by a Monte Carlo propagation of light bundles from each voxel. As these light bundles are propagated across the grid, light is deposited in traversed voxels in proportion to the calculated cross section and path length in that voxel. Any light deposited is then subtracted from the origin voxel. At the end of each iteration, each voxel contains an array describing the angular distribution of the light that escapes the grid, as well as another array that contains light deposited from scattering light originating in other voxels. This second array of light newly deposited in the voxel is then used for any subsequent multiple scattering iterations.

At the end of all the Monte Carlo multiple scattering interations, the various arrays with the light values arc reduced to a single array containing the directionally dependent luminosity of that voxel as seen from outside the grid. In order to calculate tile correct illumination for view points that arc inside the grid, this luminosity is corrected for the directionally dependent attenuation between each voxel and the edge of the grid in that direction.

Two implementations of the Monte Carlo multiple scattering code were developed; a distributed processing version implemented in MPI [7], running on a cluster of workstations and 4 processor mini-computers, and a threaded version running 011 a 32 processor SGI shared memory machine.

4.1. Multiple Scattering via distributed processing

Since the single scattering phase function is peaked strongly in the forward direction[8], the grid data is distributed to each processor in strips along the sunlight direction. Light bundles that cross to another processor's data domain are passed to that processor in an MPI data structure. The number of interprocess communications increase rapidly as a function of the number of processors. Comets have a head & tail structure requiring that the grids have an aspect ratio of 2:1 or more. As the aspect ratio increases, the fraction of total photons (messages) that cross from one processor region to another increases rapidly. Figure 1 illustrates this, with the horizontal division representing regions of the grid allocated to two different processors, anc1 the small boxes representing voxels within that region.

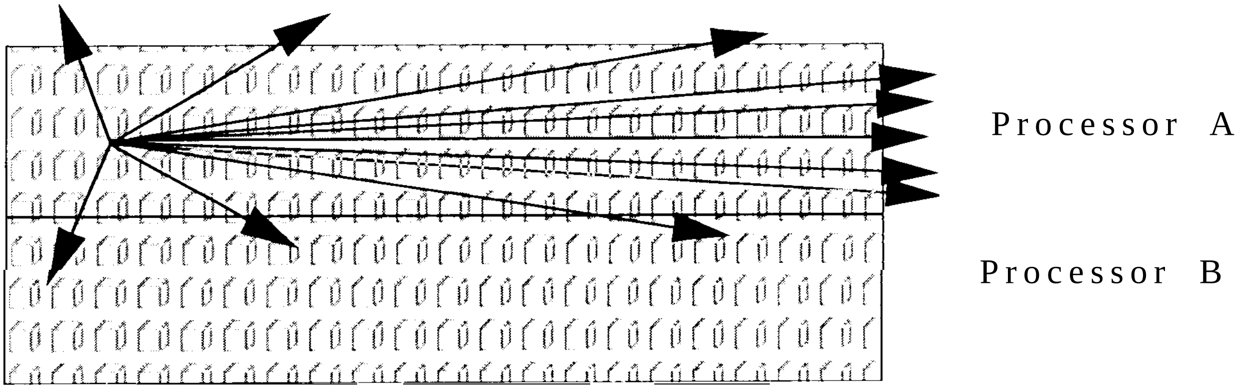

Figure 1. Schematic of scattered light propagation. The light scattering is dominantly in the forward direction (to the right), so the grid is distributed to the processors in strips along the initial light propagation direction.

The Cartesian axes are defined such that the initial plane parallel sunlight travels in the +z direction. During the propagation of scattered light, light bundles are passed to nearest neighbor processors along the x and y axis, if they cross the grid boundaries assigned to each processor. These light bundles are buffered, and passed along once 100 of them accumulate. These passed light bundles are received and processed, and the resulting photons are either passed along the next processor, or escape the edge of the grid. Each processing of passed photon bundles produces an equivalent sized buffer of accounting data that is broadcast to all processors. Though the volume of accounting data broadcast by each processor is linear with the total number of processors, these accounting packets are computationally inexpensive compared with the propagation of the photon bundles.

For the following table, every voxel in the grid had 100 direction bins, with a total of 1000 photon bundles generated for each voxel. Three SGI Onyx machines, each with 4 processors were used, a single processor SGI Impact and a dual-processor SGI Octane. All machines have similar CPU and clock speeds, with the major differences in the amount of physical memory. cf. Table 2. The non-Onyx machines were only used for the twelve processor runs. The Onyx machines MPI package uses a shard memory message passing mechanism where possible, greatly increasing the speed of communication between processes resident on the same machine. The inclusion of the Impact and Octane resulted in increase of computation time rather than a decrease, possibly because of the increased communication time. Additionally, these machines have a much smaller amount of physical memory, and may have been paging, thus greatly effecting the efficiency of these runs.

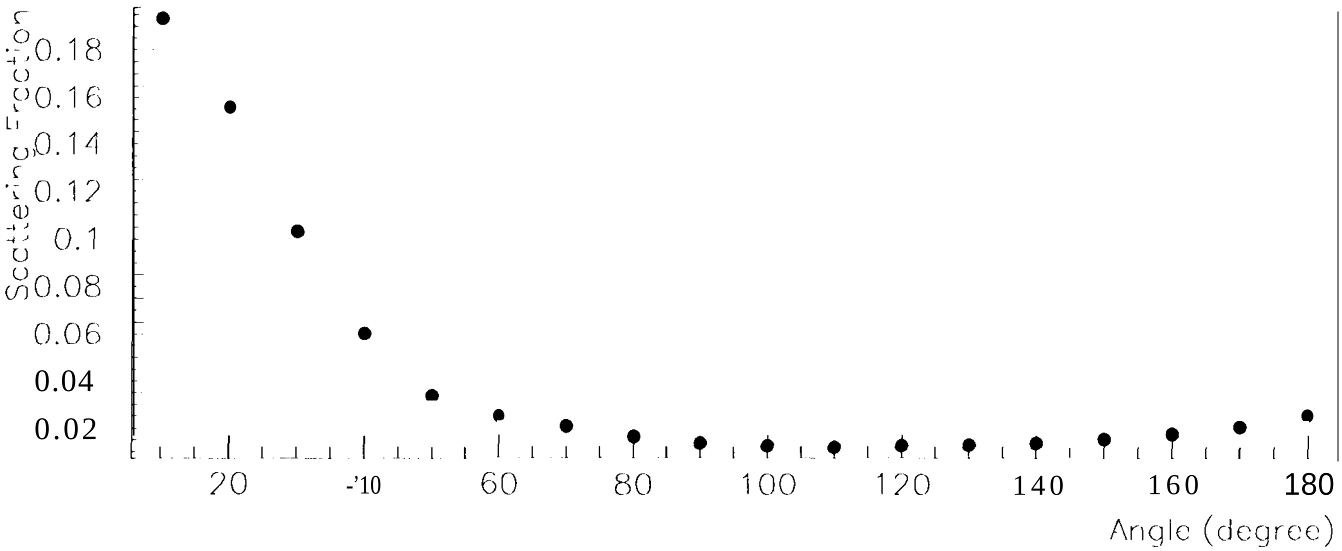

Figure 2. Devine-Ney scattering amplitude as a function of angle.

4.2. Threaded Multiple **Scattering**

All of the threaded runs were performed 011 an 32 node SGI Origin with 8 GB of physical memory. Each thread was assigned a set of voxels, selected by their index in the grid. Though the index advanced most rapidly in z, this scheme does not fully isolate each thread in a region along the z-axis. In the cases where a thread propagates a light bundle into a region assigned to another thread, there is a probability that either of these threads may block 011 a lock placed by the other thread 011 access to the various light arrays. As the number of threads increases, the number of collisions 011 these locks will increase. Thus the non linear decrease of computation time with increasing number of processors in 'J'able 1 is expected.

Table 1
Processing time M 1'1 vs. Threaded

# Voxels	# of processors	MPI	Threaded
4096	2	2020	630 s
4096	4	900	310 s
4096	9	690	130 s
4096	12	1025	93 s
13824	2	14320 *	1840 s
13824	4	6270	1190 s
13824	9	3485	510 s
13824	12	5340 *	390 s
216000	32		.5400 s

* These runs were done with 100 photon bundles/voxel, with the time adjusted by a factor of 1 O. Previous testing has indicated that the processing time is approximately linear with the number of photon bundles/voxel used.

[i

5. Rendering: Distributed vs. Threaded

Code to rendering images of the the generated comet coma has been developed using both the MPI-distributed and threaded methods. In both the threaded and Nil]] distributed implementations, the imaging plane is divided into a N x M sub regions, each task given to a slave processor or thread. There is no inter-thread or inter-slave communication. The processing time for the MPI distributed and threaded versions of the renderer are similar. The major difference in rendering time is due to the overhead of partitioning and sending the grid to each processor in the MPI distributed implimentation. Currently, the MPI distributed version of the renderer has a master that extracts and transmits the portions of the grid needed by each slave processor. In future versions of the MPI distributed renderer, each slave processor will strip the voxels in it's field of view fr'0111 a single or distributed data file, significantly increaseing the rendering time.

Figure 3 is a rendering of a synthetic comet generated using the model discussed in this paper.

6. Discussion

For the multiple scattering code, the buffering size of 100 photons (80K bytes) used to produce the benchmarks above is very likely non optimal. Attempt S to use very large (80 M bytes) buffers to minimize the number of communications produced a dramatic increase in computation time. In the other direction, buffer sizes of 8K or less resulted in failure of the MPI code, possibly due to too many unprocessed communications requests stacking up.

The large size of the data structures used (250-2000” MBytes) makes a robust implementation of the Monte Carlo multiple scattering code on a non-virtual memory system like the Cray-T3I) problematic. Though the grid can be distributed in the memory of the individual processors, the message buffers that are passed among the processors take up considerable space. Since the multiple scattering code is run infrequently, and offline, it is unlikely that any time illw’sled in porting the multiple scattering code to the T3D would be well spent.

For the rendering problem, the communications load is trivial, and memory resource problems arc tractable. An implementation of the renderer for the T3D is under development. The rendering problem is worth the additional time invested, given the frequency with which rendering is done, and the pseudo-realtime demands of spacecraft mission simulation.

Table 2
Machines used

Machine	Memory	processor
Onyx	1 GB	4R10000 195MHz
Onyx	2 GB	4 1{10000 195MHz
Onyx	2 GB	4 1110000 195MHz
Impact	128MB	1 1{10000 195MHz
Octane	128MB	2 1/10000 195MHz
Origin	8 GB	32 R10000 195MHz

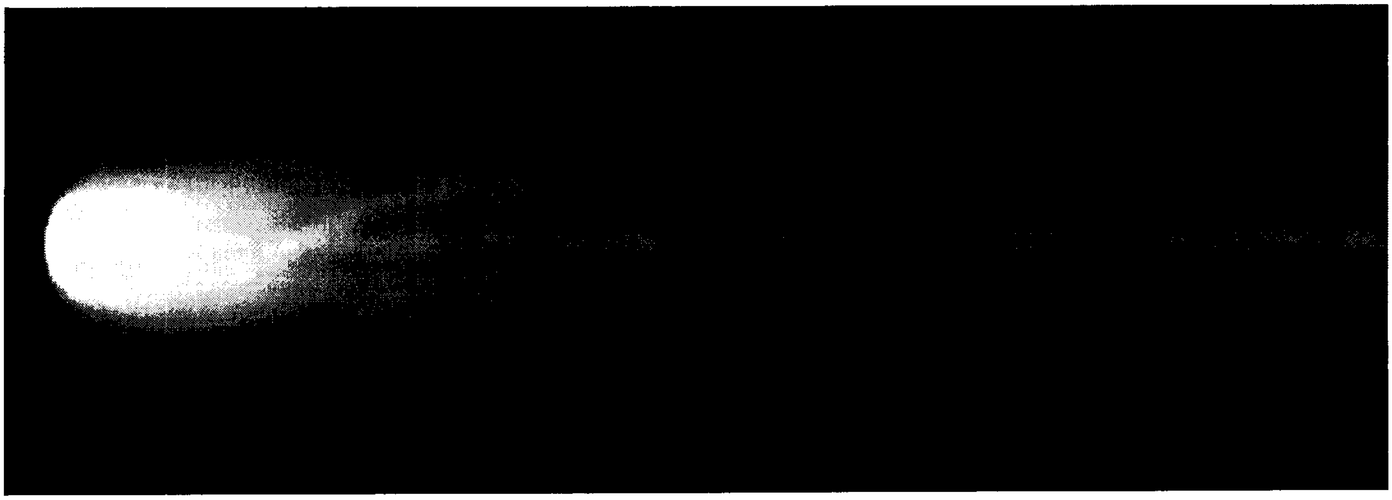

Figure 3. Image of a synthetic Halley-class comet, with a single jet and a single axis spin about the solar-comet axis.

REFERENCES

1. Nelson Max
   Efficient Light Propagation for Multiple Anisotropic Volume Scattering
   presented at The Fifth Eurographics Workshop 011 Rendering, Darmstadt, Germany, 1994.
   Optical Models for Direct Volume Rendering
   IEEE Transactions on Visualization and Computer Graphics, Vol. 1 no. 2, June 1995.
2. 1). 1111111['1, D. Greenburg, M. Cohen
   A Radiosity Method for Non-Diffuse Environments
   Computer Graphics, Vol. 20, no. 4, August 1986, pp. 133-142.
3. Tomoyuki Nishita, Yoshinori Dobashi, Eihachiro Nakamae
   Display of Clouds Taking into Account Multiple Anisotropic Scattering and Sky Light
4. s. Nappo, J. A.M. McDonnell, A. C. Levasseur-Regourd, J.C. Mandeville, A. Soubeyran, J .C. Sarnecki
   Intercomparison of Giotto DIDSY/PIA and HOPE Data.

Asteroids Comets Meteors III
Proceedings of a meeting held at the Astronomical Observatory of the Uppsala University, June 12-16 1989 pp 397

5. Bertram Donn
The Accumulation and Structure of Comets
Comets in the Post-Halley Era, Vol I, pp 335

6. David J. Lein
Optical Properties of Cometary Dust Comets in the Post-Halley Era, Vol II, pp 1005

J.A.M. McDonnel, P. L. Lamy, and G. S. Pankiewicz
Physical Properties of Cometary Dust Comets in the Post-Halley Era, Vol II, pp 1043

7. MPI, a Message Passing Interface c.f. http://www.mcs.anl.gov/mpi/index.html

8. R.L. Newburn, personal communication, and
Edward P. Ney, K.M Merrill *Comet West and the scattering function of Cometary Dust* Science Vol. 104, 3 Dec 1976, pp 1050